Autonomous Operation for Last-Mile Food, Grocery, and Goods Delivery on an Urban Sidewalk-Project Update

Keith Chester, Bob DeMont

February 21, 2022

Abstract

The purpose of the project is to simulate successful autonomous navigation of a delivery robot along sidewalks in a busy urban environment from store to a delivery address on ground level.

Background

Last mile delivery has been a growing issue with the increase in e-commerce over at least the last 8 years with each year growing from 15-30% from the year before [1]. Since much of this final delivery is accomplished via vehicles, the has been an accompanying call for reduced emissions. A McKinsey study estimates that this could lead to a 25% increase in CO2 emission in cities [2]. Most recently, the COVID pandemic with its isolation mandates have led to even greater demand for delivery of not only goods but also everyday essentials such as meals, groceries, and prescription medicine. This needed to be accomplished with minimal human contact while fewer humans were available due to quarantines and employee availability. For grocery delivery, the need for fastest route planning is also necessary due to the presence of perishables and temperature sensitive cargo - or, in other words, people prefer their meals hot and their frozen goods cold. Cheng et al [4] have shown efficient curb detection to set the limits of the robot's path.

Goals

Our goal is to create planning and execution algorithms to permit autonomous navigation and avoidance of both static (mailboxes, trashcans, signs)and dynamic obstacles (people, cars) while following sidewalk rules (crosswalks etc).

Proposed Methods

We plan to explore various navigation, collision detection and avoidance methods implemented in Python to successfully navigate around static obstacles, avoid dynamic obstacles and reach the destination in a simulated urban environment in Gazebo.

Expected Results

We expect to create a Gazebo-based simulation to navigate from the store to a nearby, ground level delivery address via sidewalks. Along the way, we plan to encounter static obstacles such as fire hydrants, garbage cans and mailboxes. We also expect to encounter moving obstacles like people, cars, or other robots. Through this we will avoid collisions to arrive at the delivery destination while following sidewalk rules.

Proposed Schedule

Our originally proposed schedule is as follows:

Milestone	Date
Proposal	Feb 1
Gazebo learning	Feb 1-18
Status Update	Feb 21
Sample environment	Feb 28
Planning algorithm	Mar 14
Obstacle avoidance algorithms	Apr 11
Final Writeup and Presentation	Apr 25-29
Final Submission	May 2

Progress

Our progress so far has consisted of a few key decisions (such as robot design) and significant efforts to familiarize ourselves with ROS tooling.

Learning ROS

We are still on track with the previously proposed schedule. We are currently inudating ourselves in the chosen tooling for our projects. Bob DeMont is currently bringing himself up to speed with Linux and Python. Both he and Keith Chester are learning the internal architecture and common development patterns

for ROS [5]. Likewise, both are learning how to design for and utilize the Gazebo simulator. We have yet to complete world development which is scheduled for within the next fortnight.

Robot Choices

We've settled on a non-holonomic 4 wheeled robot. It will utilize a GPS sensor for a pre-determined, global path plan and LIDAR for more specific localized obstacle avoidance during operation. Other sensors (such as a camera) may be included on the robot to match capabilities seen in current delivery robots, but we will likely depend on LIDAR for mapping. We will assume that the odometry measurements in the robot are without noise as we are experimenting with path planning and not sensor fusion or localization techniques.

We will utilize existing ROS modules wherever possible to handle localization, mapping, and sensor feedback. We will, however, create custom implementations of key navigation algorithms to demonstrate our competency as per the project.

Docker

During this time we also explored the use of Docker as a development tool. We began by referencing wikis maintained by the ROS organization to get our initial images configured [6]. From there we read several blogs and sources attempting to find a balance of ease of approach with industry best practices. One notable suggestion, while focused on ROS2, talked at length about portable development environments complete with all necessary pieces [7].

While we utilized resources to familiarize ourselves with the tooling necessary to be successful in a ROS project, we dutifully recreated much of what we were working with in our containerized environments. Lessons-learned were collected and offered in the class discussion forum. It is our hope that these tips and instructions aid our classmates.

Division of Labor

We plan extensive and equal collaboration on all aspects of the project to include coding, algorithm development, testing and submissions. Further delineation of work may occur as we determine proficiency in certain aspects of the project.

References

[1] https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/

[2] https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/efficient-and-sustainable-last-mile-logistics-lessons-from-japan

[3] M. Kocsis, J. Buyer, N. Sußmann, R. Zöllner and G. Mogan, "Autonomous Grocery Delivery Service in Urban Areas," 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2017, pp. 186-191, doi: 10.1109/HPCC-SmartCity-DSS.2017.24.

[4] M. Cheng, Y. Zhang, Y. Su, J. M. Alvarez and H. Kong, "Curb Detection for Road and Sidewalk Detection," in IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 10330-10342, Nov. 2018, doi: 10.1109/TVT.2018.2865836.

[5] https://www.theconstructsim.com/

[6] http://wiki.ros.org/docker/Tutorials/Docker

[7] https://www.allisonthackston.com/articles/vscode-docker-ros2.html

Appendix-Contributions

Keith Chester- After establishing best practices for ROS within Docker for rapid development and documenting it for the team, I have also been working hard at learning the internals of the ROS Navigation Stack, URDF/XARCO modeling for Gazebo, and Gazebo world creation. I will be creating the URDF for the robot model and working on the city street world we'll need for the final project simulation.

Bob DeMont- I have spent the bulk of the time learning the tools for the project. I have a setup and functioning Linux virtual machine with ROS and Gazebo working. I am new to Linux, Python and, of course, ROS and Gazebo. I have progressed through section 5 of **The Construct** courses, read a good portion of the ROS wiki and viewed virtually every video I can find involving ROS and Gazebo. I'm able to create packages, nodes, topics and publish and subscribe. In the next week I plan to study world development and begin to put together a world environment for the simulation involving a sidewalk along a building with various stationary obstacles including a dumpster, cone and mailbox. Following that I'll learn to introduce dynamic obstacles .